DOI: https://doi.org/10.52756/ijerr.2024.v46.027 *Int. J. Exp. Res. Rev.*, Vol. 46: 342-354 (2024)

Original Article | Peer Reviewed | Open Access

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in

# An Elitism-based Novel Approach for Community Detection in Social Networks

**Ranjana Sikarwar[1]*, Shyam Sunder Gupta[1] and Harish Kumar Shakya[2]**

[1]Department of Computer Science and Engineering, Amity University, Gwalior, Madhya Pradesh, India; [2]Department of Artificial Intelligence & Machine Learning, Manipal University, Jaipur, Rajasthan, India

**E-mail/Orcid Id:**

*RS*, ranjana.sik@gmail.com, https://orcid.org/0000-0002-9640-1145; *SSG*, ssgupta@gwa.amity.edu, https://orcid.org/0000-0001-6841-3897; *HKS*, harish.shakya@jaipur.manipal.edu, https://orcid.org/0000-0002-5401-3507

**Article History:**

**Received:** 24th Aug., 2024
**Accepted:** 25th Dec., 2024
**Published:** 30th Dec., 2024

**Keywords:**

Genetic algorithm, Elitism, Modularity, Community detection, Social networks, Multi-objective, Swarm-intelligent techniques, Convergence, Local optima

**How to cite this Article:**

Ranjana Sikarwar, Shyam Sunder Gupta and Harish Kumar Shakya (2024). An Elitism-based Novel Approach for Community Detection in Social Networks. *International Journal of Experimental Research and Review*, *46*, 342-354.
**DOI**: https://doi.org/10.52756/ijerr.2024.v46.027

**Abstract:** The detection of communities is an important problem in social network analysis, which has applications in various domains like sociology, biology, computer science, and marketing. In this context, genetic algorithms have proven to be effective in detecting communities by optimizing the modularity score of the network. The proposed work in this research paper uses an elitism-based genetic algorithm with some modified crossover and mutation techniques to detect communities in social networks. The proposed methodology incorporates the concepts of elitism, N-point crossover, and inverse mutation to enhance the effectiveness of genetic algorithms in solving optimization problems. The idea introduced in this article significantly extends the current understanding of optimization and evolutionary algorithms. We present an advanced methodology that leverages various genetic operators to improve the performance of a genetic algorithm in solving community detection problems in complex networks. Numerous research papers have extensively showcased the practicality of evolutionary and swarm-based algorithms in addressing real-world problems across diverse domains like viral marketing, link prediction, influence maximization, political polarization, etc. Hybridizing these algorithms with other optimization techniques has improved the performance and convergence speed, leading to enhanced optimization outcomes.

## Introduction

Social network analysis (SNA) is used to study social structures by analyzing the relationships between individuals and communities. The domain of SNA combines theory, methods, and computer technology to explore social networks and their interaction patterns using network and graph theory. The study observed it in the form of networked structures with nodes as users, objects, or things acting in a network and edges as relationships connecting these nodes (Fortunato, 2010; Pizzuti, 2018). SNA offers tools for analyzing social networks, identifying key factors and relationships, and studying the dynamics of social processes. It is used in various fields, such as sociology, anthropology, psychology, organizational studies, and computer science, for observing social influence, cooperation, diffusion of innovations, social support, and disease transmission.

Social networks play a pivotal role in driving social change by enabling the rapid spread of ideas, collective action, and mobilization. They offer platforms for individuals to share information, voice concerns, and organize movements, fostering awareness and activism. For example, social networks have been instrumental in movements like the Arab Spring, Black Lives Matter, and environmental protests, as they allow people to connect across borders, coordinate activities, and amplify voices that may otherwise be marginalized. The ability to rapidly disseminate information and rally support through online platforms has reshaped how social change occurs, making it more global and participatory (Camacho et al., 2020).

***Corresponding Author:** harish.shakya@jaipur.manipal.edu

Social networks significantly influence individual behavior by shaping opinions, preferences, and actions through peer pressure, social norms, and exposure to diverse perspectives. The content individuals consume on social media platforms can impact their attitudes toward various topics such as politics, consumer choices, and lifestyle. For instance, the "social proof" effect suggests that people tend to adopt behaviors they observe in others, such as purchasing trends, political affiliations, or lifestyle changes. Social networks also provide feedback loops where individuals' behaviors are reinforced by likes, comments, or shares, further driving conformity to certain norms or behaviors (A.J. and P.D., 2015).

Social networks are powerful tools for spreading news, information, and ideas due to their speed, reach, and viral potential. Content shared on social media platforms can quickly go viral, reaching millions of people in a short time. This spread is often amplified by algorithms that prioritize engaging content, making it more likely that sensational or highly engaging news and ideas are circulated. The decentralized nature of social networks allows individuals to create and distribute information, bypassing traditional gatekeepers like journalists or publishers. However, this also means misinformation and fake news can spread just as rapidly, creating challenges in ensuring accuracy and credibility (Che et al., 2021).

Social networks influence organizations and institutions by changing how they communicate, collaborate, and interact with stakeholders. Many businesses and governmental institutions now rely on social media to engage with customers, gather feedback, and promote their services. Social networks also enable greater transparency and accountability, as consumers and citizens can voice their opinions publicly. Social media platforms have also transformed marketing and public relations strategies, making them more interactive and direct. However, organizations must also navigate challenges such as managing public relations crises, dealing with negative publicity, and addressing the spread of misinformation that can affect their reputation and operations (Osaba et al., 2018; Lancichinetti et al., 2008). Social network analysis (SNA) is a powerful tool for understanding the community world as it relates to social ties. It can be used to identify patterns of social interaction, measure the strength of social ties, and track the flow of information and resources through social networks. Moreover, SNA can predict future behavior and design interventions to improve social outcomes (Frey, 2022; Borgatti, 2005).

The detection of communities in social networks is the task of identifying groups of nodes (called communities or modules) where nodes within a community are more tightly connected than nodes outside it. This process aims to uncover the underlying structure or organization of the network by identifying these communities. Several methods for detecting community structures are extensively reviewed and discussed (Fortunato, 2010).

#The detection of communities in social networks involves various methods, each tailored to address specific network structures and complexities. Modularity Maximization seeks to identify a division that maximizes modularity, quantifying how closely connected nodes within a community are compared to their external connections. This method is especially effective for networks with a clear community structure (Fortunato, 2010).

#Label Propagation assigns labels iteratively to nodes, clustering those with similar connections under the same label. This technique adapts well to dynamic or ambiguous networks. Spectral Clustering, leveraging eigenvalues and eigenvectors from the network's adjacency matrix, provides a mathematical basis for detecting and characterizing communities (Fortunato, 2010).

# Evolutionary algorithms further enhance community detection by employing natural evolution-inspired principles to optimize the solution space. Techniques like Genetic Algorithms (GA) iteratively evolve populations of potential solutions using genetic operators like mutation and crossover (Pizzuti, 2009; Guo et al., 2019; Panizo-Lledot et al., 2020). Similarly, Particle Swarm Optimization (PSO) mimics flocking behavior, allowing solutions to converge through collective movement towards optimal solutions (Rahimi et al., 2017).

Other algorithms, including Ant Colony Optimization (ACO) (He et al., 2011), Differential Evolution (DE) (Jia et al., 2012; Xiao et al., 2018) and Artificial Bee Colony (ABC) (Wang et al., 2016; Hafez et al., 2014; Che et al., 2021), simulate biological behaviors to identify well-separated and dense communities in large networks.

By applying these techniques, researchers can uncover the dynamics of social systems, such as how social ties bind communities, the flow of information, and the factors influencing social changes. Community detection is pivotal in domains like social network analysis, biology, and computer science, enabling insights into complex system structures (Fortunato, 2010; Pizzuti, 2018).

DOI: https://doi.org/10.52756/ijerr.2024.v46.027

## Advantages of Evolutionary Algorithms in Community Detection

**Ability to Detect Dense and Well-Separated Communities:** Evolutionary algorithms excel at identifying communities that are both tightly interconnected internally and distinctly separated from other communities. This ensures high-quality clustering in complex networks (Pizzuti, 2018; Guo et al., 2019).

**Scalability:** These algorithms effectively handle networks with a large number of nodes, maintaining efficiency and accuracy even as network size increases (He et al., 2011; Jia et al., 2012).

**Ease of Implementation:** Evolutionary algorithms are relatively straightforward to implement, making them accessible for researchers and practitioners across various domains. Their adaptability allows for customization to meet specific network analysis requirements (Hafez et al., 2014; Che et al., 2021).

Previous studies by researchers have utilized various encoding schemes and genetic operators with diverse parameter settings for solving problems in complex networks. A simple genetic algorithm with genetic operators is depicted in Figure 1. Several enhancements have been proposed to improve the quality of population generation in genetic algorithms. One such enhancement is elitism, a selection strategy in genetic algorithms designed to retain the best-performing individuals in each generation. This technique ensures that the highest-quality solutions are preserved for the subsequent generation without any modifications (Sharma and Shakya, 2022).

Elitism is fundamentally based on the idea that certain individuals or groups in a population are superior to others. Optimization involves transferring the best solutions from one generation to the next to avoid their loss during the evolutionary process (Sharma et al., 2023).

## Advantages of Elitism

Preservation of High-Quality Solutions: By retaining the top-performing individuals, the algorithm prevents the loss of promising solutions and maintains a consistently high-quality population (Sharma et al., 2024a). Avoidance of Local Optima: Elitism ensures that the search is not trapped in local optima, facilitating convergence toward globally optimal solutions (Sharma et al., 2024b). Faster Convergence: Particularly in small population sizes, elitism expedites the convergence process by focusing on refining the best solutions (Sharma and Shakya, 2024).

## Implementation of Elitism:

Elitism is implemented by selecting a predetermined number of top-performing individuals from the current population and transferring them directly to the next generation. The remaining individuals are then chosen using other strategies such as the roulette wheel or tournament selection. Elitism balances exploiting existing solutions with exploring new possibilities (Sharma et al., 2024c).

**#Non-strict Elitism:** In this technique, the fittest individuals are guaranteed to be carried out to the next generation, but other individuals may also be carried over if they are close enough in fitness to the fittest individuals to maintain a diverse population and avoid premature convergence.

**#Strict Elitism:** In this technique, only the superior individuals are copied to the next generation, and the others are discarded. This increases the convergence rate but can lead to premature convergence and loss of diversity.

## Elitism in Genetic Algorithms

The primary objective of elitism in genetic algorithms is to **prevent the loss of high-quality solutions** identified in previous generations. Elitism involves directly copying a subset of top-performing individuals from the current population to the next generation without applying genetic operators like crossover or mutation. This ensures that these superior solutions are preserved, while the rest of the population is selected using other strategies, such as roulette wheel selection or tournament selection (Sharma and Shakya, 2024).

## Advantages of Elitism

#**Preservation of Good Solutions**: Elitism reduces the risk of losing optimal solutions, ensuring that the best-performing individuals are carried over to subsequent generations (Sharma and Shakya, 2022).

#**Faster Convergence**: It accelerates the algorithm's convergence towards an optimal solution by retaining superior individuals (Sharma et al., 2023).

#**Improved Solution Quality**: The consistent presence of high-quality solutions ensures better population quality over generations, leading to enhanced optimization outcomes (Sharma et al., 2024a).

#**Robustness**: By preserving the best solutions, elitism increases resilience against challenges like noisy or deceptive fitness landscapes, making the algorithm more robust (Sharma et al., 2024b).

## Potential Challenges

While elitism can enhance performance, excessive use might lead to premature convergence, where the population loses diversity, resulting in suboptimal

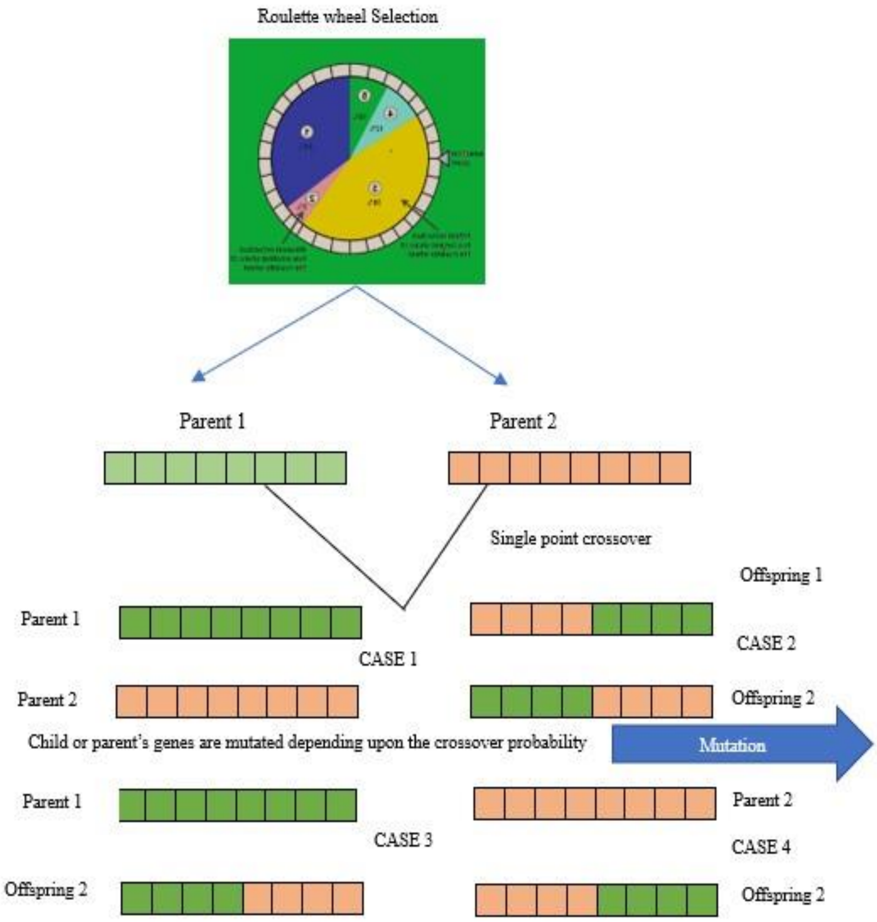


**Figure 1. Genetic algorithm working with genetic operators.**

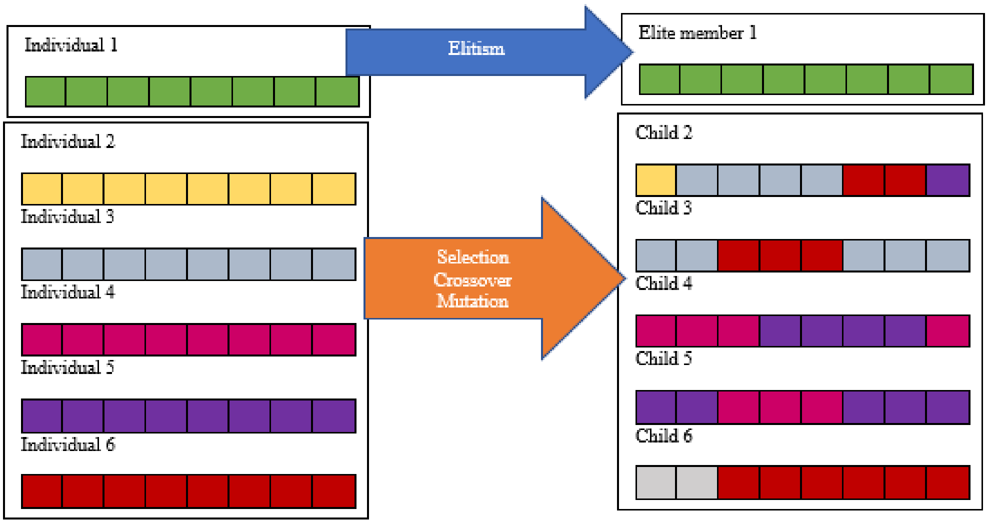


**Figure 2. Population-preserving Elite Member.**

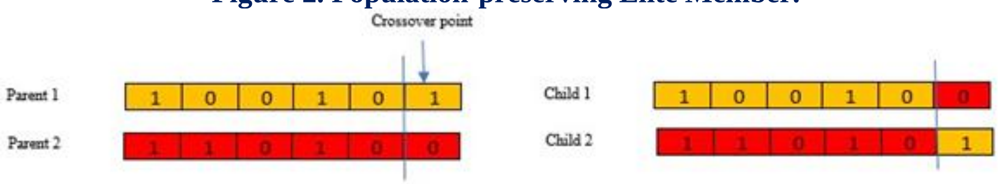


**Figure 3. One-Point crossover.**

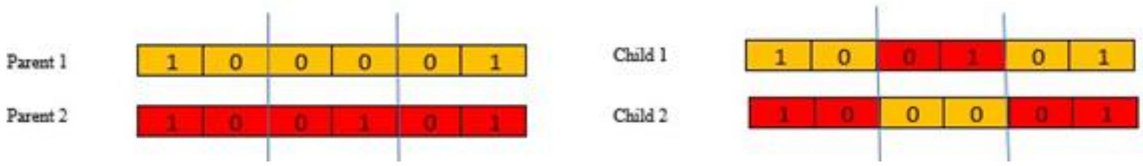


**Figure 4. Two-Point crossover.**

solutions. Balancing elitism with other selection strategies and genetic operators is crucial for maintaining a diverse and effective search space (Sharma et al., 2024b).

Elitism can be used in any application area where the genetic algorithm is applied to solve optimization problems. Some of the application areas are:

#Engineering design optimization
#Financial portfolio optimization
#Image processing and computer vision
#Robotics and control systems
#Transportation and logistics optimization
#Data mining and machine learning
#Bioinformatics and genomics
#Game theory and decision-making
#Signal processing and pattern recognition
#Simulation and modeling

Genetic algorithms can be used to find a good network partitioning by optimizing a fitness function that measures the quality of the community structure. Elitism can be used in community detection problems to preserve the best solutions found so far, such as partitioning with the highest fitness value and carrying them over to the next generation.

## Mutation Techniques

Genetic algorithms (GAs) are widely used for community detection in complex networks because they can explore large search spaces and optimize modularity functions. Several studies have applied GAs to community detection, including those by Sarkar et al. (2016), Banerjee et al. (2017) and Sanyal and Das (2021). Kar et al. (2022) proposed a hybrid GA approach that uses link strength to improve community detection accuracy. Chen et al. (2012) employed ant colony optimization (ACO) for detecting community structures, while Liu and Liu (2018) utilized differential evolution for the same purpose. Furthermore, Chen et al. (2017) enhanced the particle swarm optimization (PSO) model with the Physarum model for better detection performance. Zhang et al. (2020) introduced a whale optimization-based community detection algorithm, and Guo et al. (2017) applied the artificial bee colony (ABC) algorithm to uncover communities in complex networks. Recent developments have also seen the application of GAs in group recommender systems, such as the work by Krouska et al. (2023) and advances in metaheuristic optimization techniques for overlapping communities. Additionally, Küçük (2023) introduced a sine wave optimization technique for elitism-based search algorithms and Triwiyatno and Setiawan (2023) focused on optimizing wind farm yaw offset angles using GAs. A comparative study on the use of GAs and reinforcement learning for solving problems like the traveling salesman problem was presented by Uthayasuriyan et al. (2023). Guariso and Sangiorgio (2020) discussed improvements in multiobjective GAs and Zhou et al. (2023) explored reinforcement learning in network analysis. Huang et al. (2023) proposed a trajectory-based genetic algorithm for solving scheduling problems in complex systems.

## Literature review

Community detection in complex networks has been extensively explored using various optimization techniques. The fast-unfolding method for community detection, as proposed by Blondel et al. (2008), has become popular due to its efficiency in large networks. This approach significantly enhances the speed of community detection by optimizing modularity, which is a key metric in evaluating network partition quality (Blondel et al., 2008). Genetic algorithms (GA) have also been employed in community detection, offering a flexible and effective method for exploring network structures (Tasgin and Bingol, 2006; Malhotra, 2021). These algorithms are particularly useful in detecting communities by evolving solutions over generations, which aids in finding optimal partitions.

Ant colony optimization (ACO) has been applied to community detection with promising results. Chen et al. (2012) demonstrated that ACO can efficiently detect community structures by mimicking the pheromone-based communication between ants, which allows the algorithm to adapt to complex network topologies. Similarly, Liu and Liu (2018) utilized differential evolution (DE), which optimizes modularity density to detect communities. Their method showed that DE could enhance community detection by exploiting the modularity of network structures (Liu and Liu, 2018). Particle swarm optimization (PSO) has also been integrated into community detection algorithms, with an enhanced PSO method based on the Physarum model providing effective results for community detection in large networks (Chen et al., 2017).

Whale optimization algorithms (WOA) have emerged as another promising approach for community detection, as Zhang et al. (2020) demonstrated. WOA's social behavior-based search mechanism allows for the detection of overlapping communities, which is a critical challenge in many real-world networks. Similarly, Guo et al. (2017) explored the use of artificial bee colony (ABC) algorithms for community detection, highlighting its heuristic nature as beneficial for uncovering hidden network structures. The flexibility of ABC algorithms has

DOI: https://doi.org/10.52756/ijerr.2024.v46.027

shown great potential in detecting communities across various types of networks.

Additionally, the use of genetic algorithms has been extended to group recommendation systems, where (Krouska et al. (2023) customized grouping through genetic algorithms, demonstrating its effectiveness in domain-independent decision support systems. These advancements reflect the growing trend of applying metaheuristic optimization algorithms, such as GAs, PSO, and WOA, for both community detection and recommendation system improvements (Krouska et al., 2023). Thus, the integration of different optimization techniques continues to advance the field of community detection, enabling more efficient and accurate results across a wide range of network types.

Table 1 comprehensively summarizes key studies focused on community detection using genetic algorithms (GAs). It highlights various approaches and methodologies employed to optimize the process of identifying community structures within complex networks. The table includes an analysis of the algorithms' performance, strengths, and limitations, offering insights into the evolution of genetic algorithm-based techniques in this domain. The reviewed studies demonstrate how GAs, often combined with other optimization strategies, have been successfully applied to uncover hidden patterns and enhance the accuracy of community detection in diverse network types.

**Table 1. Literature Overview: Genetic Algorithm-based Community Detection.**

| Reference | Methodology/Approach | Key Findings | Application |
|---|---|---|---|
| Blondel et al. (2008) | Fast unfolding of communities | Proposed an efficient algorithm for community detection by optimizing modularity. | Large-scale network community detection |
| Tasgin and Bingol (2006) | Genetic Algorithm (GA) | Employed GA to detect communities in complex networks, offering flexibility and adaptability. | Community detection in complex networks |
| Malhotra (2021) | Link Strength-Based Hybrid Genetic Algorithm | Hybrid approach combining GA with link strength for community detection. | Optimized community detection using link strength-based GA |
| Chen et al. (2012) | Ant Colony Optimization (ACO) | ACO efficiently detects community structures by simulating pheromone-based communication. | Community detection in networks |
| Liu and Liu (2018) | Differential Evolution (DE) | DE optimized modularity density for community detection in complex networks. | Community detection based on modularity density |
| Chen et al. (2017) | Particle Swarm Optimization (PSO) with Physarum model | Enhanced PSO model based on Physarum for effective community detection in large networks. | Community detection using PSO and Physarum model |
| Zhang et al. (2020) | Whale Optimization Algorithm (WOA) | WOA effectively detects overlapping communities in networks. | Community detection in networks with overlapping communities |
| Guo et al. (2017) | Artificial Bee Colony (ABC) Algorithm | ABC algorithm is heuristic-based and effective for uncovering hidden network communities. | Heuristic-based community detection |
| Krouska et al. (2023) | Genetic Algorithm for Group Recommendation | Customized grouping through GA for domain-independent decision support in recommendation systems. | Group recommendation system using GA |

## Materials and Methods

In this methodology, we have used the 'elitism' parameter, which determines whether the top solution from the previous generation is transferred to another generation. We have also used N-point crossover and inverse mutation, respectively. In N-point crossover, we randomly select 'n-crossover points' positions in the parents' genomes and exchange the corresponding segments to create two offspring. In inverse mutation, we randomly select and flip a subset of the bits in the offspring genome.

### Encoding scheme

The binary encoding scheme is used to represent chromosomes in the proposed algorithm. This encoding method allows for a simple representation of the problem and enables easy mutation and crossover operations to be performed on the chromosomes. For example, in the case of a community detection problem, a binary string represents the membership of each node in the network to a particular community. Each gene in the chromosome represents a node, and the gene's value corresponds to the node's membership to a particular community. The objective function can be defined based on the modularity score, which is calculated based on the assignment of nodes to different communities. By using binary encoding, the genetic algorithm can efficiently search for the optimal assignment of nodes to communities that maximizes the modularity score.

Exploitation is the process of intensifying the search for the current best solution to improve it further. While exploration refers to the process of searching in new and unexplored search space to find potentially better solutions. Exploitation is achieved using elitism, which ensures that the best solutions found so far are always carried over to the next generation. This allows the algorithm to focus on improving the best solutions, and gradually converge towards an optimal solution.

Exploration is achieved using mutation, which introduces random changes to the solutions.

### Crossover and Mutation Techniques

N-point crossover and inverse mutation are two specific techniques used in this algorithm to balance the exploitation and exploration. N-point crossover ensures that the offspring inherit good characteristics from both parents, while inverse mutation helps to introduce more diversity into the population.

Here are the parameter values used in the above code:

- Population size:50
- Number of generations:200
- Crossover probability:0.8
- Mutation probability:0.2
- Number of crossover points (N-point crossover):2
- Number of individuals to select for tournament selection
- Inverse mutation probability:0.1

Modularity is a quality metric that measures the degree to which a network is partitioned into dense communities that are well-separated from one another. It can be calculated as a Q value using equation 1.

$$Q = \frac{1}{2m}\sum_{ij}\left[a_{ij} - \frac{k_i k_j}{2m}\right]\delta\left(c_i, c_j\right) \qquad (1)$$

Here is the pseudocode of the proposed genetic using elitism algorithm for community detection in networks:

**Inputs:**

- Network adjacency matrix
- Population size (number of individuals in the population)
- Number of generations
- Mutation probability
- Crossover probability
- Number of crossover points
- Elitism ratio

**Outputs:**

- Best partition found
- Modularity score of the best partition

**Algorithm:**

1. Generate an initial population of partitions randomly.
2. Evaluate the fitness of each individual(partition) using modularity
3. For each generation:
    a. Select parents for mating using roulette wheel selection.
    b. Perform crossover to create offspring using N-point crossover.
    c. Perform mutation on the offspring with a given mutation probability.
    d. Evaluate the fitness of the offspring.
    e. Select the best individuals to survive using elitism.
    f. Replace the worst individuals in the population with the offspring.
4. Select the best partition found in the last generation and return it along with its modularity score.

### Proposed Algorithm EBGA with different parameters

**Input**: Network G = (V, E), Population size Pop_size, Maximum number of generations max_gen, Crossover probability pc, Mutation probability pm, Elitism ratio er, Number of communities k

1. Initialize the population P with random partitions of the network into k communities
2. Evaluate the fitness of each individual in P using the modularity score
3. Sort the population in decreasing order of fitness
4. Initialize the current generation gen = 1
5. Repeat until convergence or the maximum number of generations is reached:

DOI: https://doi.org/10.52756/ijerr.2024.v46.027

a. Select two parents from the population using the roulette wheel selection
b. With probability pc, apply n-point crossover to produce two offspring
c. With probability pm, apply inverse mutation to each offspring
d. Evaluate the fitness of the offspring
e. Replace the two worst individuals in P with the two offspring if their fitness is higher
f. Sort the population in decreasing order of fitness
g. Apply elitism by replacing the worst individual in P with the best individual from the previous generation if their fitness is higher
h. Increment the generation counter gen

6. Output the best individual from P

## Result and Discussion

The algorithms discussed above, along with our proposed idea of using elitism with different crossover and mutation techniques, have been tested one by one multiple times. These experiments have been conducted on the Windows 10 operating system, on Intel(R) Core (TM) i3-7020U CPU 2.30 GHz processor with 8GB RAM using the programming language Python 3.10.0 for running small to medium-sized datasets. However, for large-size datasets more powerful hardware, such as a server with multiple CPUs or GPUs may be required to achieve reasonable performance. A series of experiments have been performed using elitism in genetic algorithm with different combinations of crossover and mutation techniques and recorded the modularity values as shown in Table 2, 3 and 4 on different-size datasets. Modularity is taken as an evaluation metric here. Initial parameter settings are shown in Table 2 and can be altered to achieve the optimal values. The algorithm was executed for 50 iterations. As referred from Table 3 where the EBGA initially uses single-point crossover and random mutation operators, the modularity values are good for Dolphin network and higher for synthetic network datasets LFR1 followed by LFR2, which shows that our algorithm performs well for larger datasets also. In Table 4 the modularity values are higher for Dolphin network and LFR1 dataset using the elitism-based genetic algorithm and a combination of N-point crossover and inverse mutation. Initially, we combined single-point crossover with random mutation on different datasets and checked for the modularity values, as shown in Table 2. After that, we tested our algorithm for 2-point, 3-point and 4-point crossover with inverse mutation and observed the results shown in Tables 4 and 5. Considering convergence criteria, the proposed algorithm converged on the 20$^{th}$ iteration for single-point crossover and on 14$^{th}$ iteration for the karate club dataset with N-point crossover and inverse mutation, respectively. Similarly, the algorithm converged on 7$^{th}$ iteration only for the American football network using N-point crossover and inverse mutation. We have tested our algorithm with different versions of GA as GA-S[40], MCOBGA, Elitism-Based Genetic Algorithm(EBGA), Modified Crossover Elitism-Based Genetic Algorithm(MCEBGA). While experimenting for MCEBGA from GA-S we applied the elitism technique to improve the modularity function Q-value and compared the results on different datasets as shown in Table 6. In the genetic algorithm with elitism implemented in this work, the best individuals from the combined population of parents and offspring are selected and preserved in the elite population. The remaining population is then filled with randomly selected individuals from the combined population. This ensures that the best individuals are preserved, and the population's overall quality is improved, leading to faster convergence.

**Table 2. Parameters of EBGA.**

| Parameter | Value |
|---|---|
| Num_gen | 50 |
| Pop_size | 100 |
| Crossover_rate | 0.8 |
| Mutation_rate | 0.2 |
| Elitism rate | 0.1 |

**Table 3. Single -point Crossover operator and Random mutation.**

| Datasets | Modularity score |
|---|---|
| Strike | 0.3092 |
| Zachary’s karate club | 0.4345 |
| American Football Network | 0.5894 |
| Bottlenose Dolphin Network | 0.4253 |
| Krebs Books About US Politics | 0.4553 |
| LFR1 | 0.6407 |
| LFR2 | 0.6691 |

**Table 4. Results with 3-point and 4-point crossover with population size=100, Generations=100, mutation rate =0.1.**

| Datasets | Number of communities | Crossover operator | Modularity score |
|---|---|---|---|
| Zachary's karate club | 2 | N-point (3) | 0.406 |
| | | N-point (4) | 0.418 |
| American Football Network | 12 | N-point (3) | 0.5515 |
| | | N-point (4) | 0.5721 |
| Bottlenose Dolphin Network | 2 | N-point (3) | 0.7788 |
| | | N-point (4) | 0.7957 |
| Krebs Books About US Politics | 3 | N-point (3) | 0.4234 |
| | | N-point (4) | 0.4319 |
| LFR1 | 4 | N-point (3) | 0.8127 |
| | | N-point (4) | 0.8216 |
| LFR2 | 28 | N-point (3) | 0.7001 |
| | | N-point (4) | 0.7143 |

**Table 5 Average results with N-point, inverse mutation, and Elitism.**

| Datasets | Number of nodes | Number of edges | Modularity score |
|---|---|---|---|
| Zachary's karate club | 34 | 78 | 0.357 |
| American Football Network | 115 | 613 | 0.597 |
| Dolphin Network | 62 | 159 | 0.503 |
| Krebs Books About US Politics | 105 | 441 | 0.412 |
| LFR1 | 128 | 1024 | 0.627 |
| LFR2 | 1000 | 7692 | 0.600 |

**Table 6. Average Modularity results with N-point, inverse mutation, and Elitism.**

| Datasets | Nodes | Edges | No. of communities | GA-S | OBGA | MCOBGA | EBGA | MCEBGA |
|---|---|---|---|---|---|---|---|---|
| Strike Network | 24 | 34 | 3 | 0.3025 | 0.3025 | 0.3056 | 0.3035 | 0.2950 |
| Karate Club | 34 | 78 | 2 | 0.2210 | 0.2225 | 0.2348 | 0.2218 | 0.2502 |
| Bottlenose Dolphin Network | 62 | 159 | 2 | 0.2608 | 0.2620 | 0.2717 | 0.2812 | 0.3014 |
| Football Network | 115 | 613 | 12 | 0.2683 | 0.2741 | 0.3017 | 0.4000 | 0.4123 |
| LFR1 | 128 | 1024 | 4 | 0.3065 | 0.3675 | 0.3694 | 0.3813 | 0.3912 |
| LFR2 | 1000 | 7692 | 28 | 0.3793 | 0.3063 | 0.5074 | 0.3599 | 0.5164 |

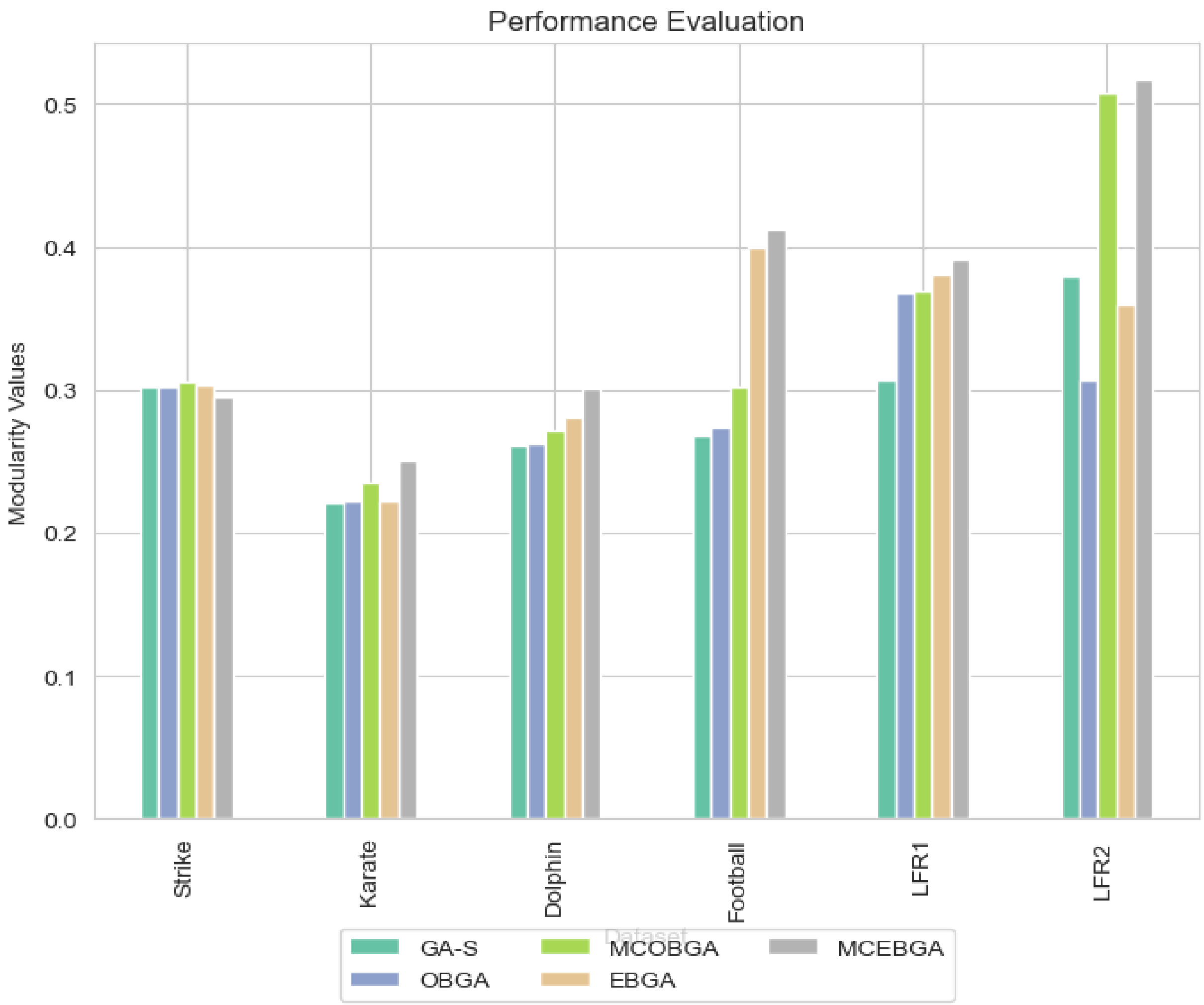


**Figure 5. Comparison graph of modularity values for different algorithms.**

## Conclusion

This research work contributes to the improvisation of the genetic algorithm by using Elitism technique in the selection phase of the genetic algorithm. According to our knowledge, the concept of elitism is used for the first time in solving community detection problems. Elitism ensures that the best individuals are not lost during the evolutionary process and that they can continue to contribute to the search for a better solution. This procedure involves transferring the top-performing individuals directly to the subsequent generation, devoid of any alteration through mutation or crossover with other individuals. Also, modularity is used as the fitness function to choose which individuals to copy to the next generation. Elitism has improved the performance of GAs in a number of ways. First, it can help to prevent the algorithm from converging prematurely on a suboptimal solution. Furthermore, this approach serves to uphold diversity within the population, thereby augmenting the algorithm's efficacy in delving into the exploration of the solution space and unearthing improved outcomes. Additionally, it contributes to expediting the algorithm's convergence process. The selection phase is typically followed by the crossover phase, in which the parents are recombined to create new offspring. We have tested our algorithm with different crossover rates and mutation rate. It is seen that a high crossover rate results in more offspring and lower mutation rate maintains the quality of the population. We have used a single-objective function for our problem using modularity. In future, we can implement it as a multi-objective function, which can further help to detect community structures with higher accuracy. Further, we tested our algorithm on small-medium-large datasets with different community structures and ground-truth communities. The algorithm showed improved performances even on large datasets also. This algorithm can be used to identify communities on social networks like Facebook, LinkedIn, Twitter, YouTube, etc., by identifying the audience or users with similar interests. Also, researchers can better understand how social networks are structured and how users interact with each other. This information can be used to improve the design of social networks, to develop new social media applications, and to understand the social world better. The proposed algorithm can be further improved by exploring various other mutation and crossover techniques. Also, different selection methods can be used to increase the diversity of the population. The algorithm

DOI: https://doi.org/10.52756/ijerr.2024.v46.027

can be improved to handle multi-objective optimization problems. Additionally, parallel computing techniques can be used to speed up the optimization process. Furthermore, the proposed algorithm can be applied to real-world problems in various fields, such as social networks, biology, and computer networks. Finally, the algorithm's performance can be compared with other state-of-the-art community detection algorithms to evaluate its effectiveness.

## Acknowledgement

Ranjana Sikarwar actively participated in the methodology, supervision, writing review and editing, research administration, experimentation, and investigation and contributed substantially to proofreading the research article. Dr. Shyam Sunder Gupta significantly contributed to various aspects, including dataset selection, algorithm design, methodology, investigation, resource management, and original draft writing. Dr. Harish Kumar Shakya played a pivotal role in overseeing, guiding, conceptualizing, and collecting references for this research endeavor.

## Conflicts of Interest

The authors assert that they do not have any conflicts of interest to disclose with respect to the current research.

## References

A.J., U., & P.D., S. (2015). Crossover operators in genetic algorithms: A review. *ICTACT Journal on Soft Computing*, *6*(1), 1083–1092. https://doi.org/10.21917/ijsc.2015.0150

Audet, C., & Hare, W. (2017). Genetic Algorithms. *Derivative-Free and Blackbox Optimization*, pp. 57–73. https://doi.org/10.1007/978-3-319-68913-5_4

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Camacho, D., Panizo-LLedot, Á., Bello-Orgaz, G., Gonzalez-Pardo, A., & Cambria, E. (2020). The four dimensions of social network analysis: An overview of research methods, applications, and software tools. *Information Fusion*, *63*, 88–120. https://doi.org/10.1016/j.inffus.2020.05.009

Che, S., Yang, W., & Wang, W. (2021). An improved artificial bee colony algorithm for community detection in bipartite networks. *IEEE Access*, *9*, 10025–10040. https://doi.org/10.1109/ACCESS.2021.3050752

Chen, Z., Liu, F., Gao, C., Li, X., & Zhang, Z. (2017). An enhanced particle swarm optimization based on Physarum model for community detection. *Lecture Notes in Computer Science*, *10386*, 99–108. https://doi.org/10.1007/978-3-319-61833-3_11

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3-5), 75-174. https://doi.org/ 10.1016/j.physrep.2009.11.002

Guariso, G., & Sangiorgio, M. (2020). Improving the performance of multiobjective genetic algorithms: An elitism-based approach. *Information*, *11*(12), 1–14. https://doi.org/10.3390/info11120587

Guo, X., Su, J., Zhou, H., Liu, C., Cao, J., & Li, L. (2019). Community detection based on genetic algorithm using local structural similarity. *IEEE Access*, *7*, 134583-134600. https://doi.org/10.1109/ACCESS.2019.2939864

Guo, Y., Li, X., Tang, Y., & Li, J. (2017). Heuristic artificial bee colony algorithm for uncovering community in complex networks. *Mathematical Problems in Engineering*, *2017*, 4143638. https://doi.org/10.1155/2017/4143638

Hafez, A. I., Zawbaa, H. M., Hassanien, A. E., & Fahmy, A. A. (2014). Networks community detection using artificial bee colony swarm optimization. *Advances in Intelligent Systems and Computing*, *303*, 229–239. https://doi.org/10.1007/978-3-319-08156-4_23

Hajipour, V., Tavana, M., Santos-arteaga, F. J., Alinezhad, A., & Di Caprio, D. (2020). An efficient controlled elitism non-dominated sorting genetic algorithm for multi-objective supplier selection under fuzziness. *Journal of Computational Design and Engineering*, pp. 469–488. https://doi.org/10.1093/jcde/qwaa039

He, D., Liu, J., Liu, D., Jin, D., & Jia, Z. (2011). Ant colony optimization for community detection in large-scale complex networks. *Proceedings of the 2011 7th International Conference on Natural Computation* (ICNC), *2*, 1151–1155. https://doi.org/10.1109/ICNC.2011.6022234

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & de Carvalho, A. C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (Applications and Reviews), *39*(2), 133–155.

https://doi.org/10.1109/TSMCC.2008.2007252

Huang, S., Tsai, Y. C., & Chou, F. D. (2023). A trajectory-based immigration strategy genetic algorithm to solve a single-machine scheduling problem with job release times and flexible preventive maintenance. *Algorithms, 16*(4). https://doi.org/10.3390/a16040207

Jaradat, A. S., & Hamad, S. B. (2018). Community structure detection using firefly algorithm. *International Journal of Applied Metaheuristic Computing*, *9*(4), 52–70. https://doi.org/10.4018/IJAMC.2018100103

Jia, G., et al. (2012). Community detection in social and biological networks using differential evolution. *Lecture Notes in Computer Science*, *7219*, 71–85. https://doi.org/10.1007/978-3-642-34413-8_6

Krouska, A., Troussas, C., & Sgouropoulou, C. (2023). A novel group recommender system for domain-independent decision support customizing a grouping genetic algorithm. *User Modeling and User-Adapted Interaction*. https://doi.org/10.1007/s11257-023-09360-3

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E, 78*(4). https://doi.org/10.1103/PhysRevE.78.046110

Liu, C., & Liu, Q. (2018). Community detection based on differential evolution using modularity density. *Information, 9*(9). https://doi.org/10.3390/info9090218

Malhotra, D. (2021). Community detection in complex networks using link strength-based hybrid genetic algorithm. *SN Computer Science*, *2*(1). https://doi.org/10.1007/s42979-020-00389-4

Natesha, B. V., & Guddeti, R. M. R. (2021). Adopting elitism-based genetic algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment. *Journal of Network and Computer Applications*, *178*, 102972. https://doi.org/10.1016/j.jnca.2020.102972

Osaba, E., et al. (2018). Community detection in weighted directed networks using nature-inspired heuristics. *Lecture Notes in Computer Science*, 11315. Springer International Publishing. https://doi.org/10.1007/978-3-030-03496-2_36

Panizo-LLedot, A., Bello-Orgaz, G., & Camacho, D. (2020). A multi-objective genetic algorithm for detecting dynamic communities using a local search driven immigrant's scheme. *Future Generation Computer Systems*, *110*, 960–975. https://doi.org/10.1016/j.future.2019.10.041

Pizzuti, C. (2008). GA-Net: A genetic algorithm for community detection in social networks. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5199 LNCS, 1081–1090. https://doi.org/10.1007/978-3-540-87700-4_107

Pizzuti, C. (2009, November). A multi-objective genetic algorithm for community detection in networks. *In 2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pp. 379-386. https://doi.org/ 10.1109/ICTAI.2009.58

Pizzuti, C. (2017). Evolutionary computation for community detection in networks: A review. *IEEE Transactions on Evolutionary Computation, 22*(3), 464-483. https://doi.org/10.1109/TEVC.2017.2737600

Putera, V. S., Permana, R. H., & Sirodj, D. A. N. (2023). Is the Indonesian Government Islamophobic? Studies Using Social Network Analysis. *KnE Social Sciences*, pp. 1397-1408. https://doi.org/10.18502/kss.v8i18.14342

Rahimi, S., Abdollahpouri, A., & Moradi, P. (2017). SC. Swarm and Evolutionary Computation. https://doi.org/10.1016/j.swevo.2017.10.009

Shakya, H. K., Singh, K., More, Y. S., & Biswas, B. (2022). Opposition-based genetic algorithm for community detection in social networks. *Proceedings of the National Academy of Sciences of India, Section A-Physical Sciences*, *92*(2), 251–263. https://doi.org/10.1007/s40010-020-00716-7

Sharma, S., & Shakya, H. K. (2022). Hybrid real-time implicit feedback SOM-based movie recommendation systems. *In International Conference on Computing, Communications, and Cyber-Security*, pp. 371–388. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1479-1_28

Sharma, S., & Shakya, H. K. (2022, October). Hybrid real-time implicit feedback SOM-based movie recommendation systems. *In International Conference on Computing, Communications, and Cyber-Security*, pp. 371–388. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1479-1_28

Sharma, S., & Shakya, H. K. (2024). Hybrid recommendation system for movies using artificial neural network. *Expert Systems with Applications*, *258*, 125194. https://doi.org/10.1016/j.eswa.2024.125194

Sharma, S., Dubey, G. P., & Shakya, H. K. (2024). Optimizing user satisfaction in movie recommendations using variable learning rates and dynamic neighborhood functions in SOMs. *International Journal of Experimental Research and Review*, *41*(spl.), 130–145. https://doi.org/10.52756/ijerr.2024.v41spl.011

Sharma, S., Dubey, G. P., & Shakya, H. K. (2024). Reducing cluster overlap in movie recommendations with IKSOM and silhouette clustering. *International Journal of Experimental Research and Review*, *42*, 169–182. https://doi.org/10.52756/ijerr.2024.v42.015

Sharma, S., Dubey, G. P., Shakya, H. K., & Motwani, D. (2023). Hybrid filtering methods in movie recommendation: The enhanced SOM approach. *In International Conference on Information Systems and Management Science*, pp. 174–187. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-70789-6_14

Sharma, S., Prasad, G., Kumar, H., & Sharma, A. (2024). SOM and hybrid filtering: Pioneering next-gen movie recommendations in the entertainment industry. *Journal of Fusion: Practical Applications*, *16*(2), 43–62. https://doi.org/10.54216/FPA.160204

Sharma, S., Shakya, H. K., & Marriboyina, V. (2021). A location-based novel recommender framework of user interest through data categorization. *Materials Today: Proceedings*, *47*, 7155–7161. https://doi.org/10.1016/j.matpr.2021.06.325

Tasgin, M., & Bingol, H. (2006). Community detection in complex networks using genetic algorithm. Retrieved from http://arxiv.org/abs/cond-mat/0604419

Triwiyatno, A., & Setiawan, I. (2023). Optimization of wind farm yaw offset angle using online genetic algorithm with a modified elitism strategy to maximize power production. *Journal of Information Technology and Electrical Engineering*, *9*(1), 185–199. https://doi.org/10.26555/jiteki.v9i1.25747

Uthayasuriyan, A., G, H. C., Uv, K., Mahitha, S. H., & Jeyakumar, G. (2023). A comparative study on genetic algorithm and reinforcement learning to solve the traveling salesman problem. *Research Article in Special Issue: Selected Papers from the 4th International Conference on Machine Learning, Image Processing, Network Security, and Data*, 1–12. https://doi.org/10.37256/rrcs.2320232642

Wang, Z., Zhao, X., Wen, P., Xue, J., & Hu, C. (2016). Community detection in complex networks using improved artificial bee colony algorithm. *Proceedings of CIMNS-16*, 283–288. https://doi.org/10.2991/cimns-16.2016.71

Xiao, J., Zhang, Y. J., & Xu, X. K. (2018). Convergence improvement of differential evolution for community detection in complex networks. *Physica A: Statistical Mechanics and Its Applications*, *503*, 762–779. https://doi.org/10.1016/j.physa.2018.02.072

Zhang, Y., Liu, Y., Li, J., Zhu, J., Yang, C., Yang, W., & Wen, C. (2020). WOCDA: A whale optimization-based community detection algorithm. *Physica A: Statistical Mechanics and Its Applications*, *539*, 122937. https://doi.org/10.1016/j.physa.2019.122937

Zhou, D., Du, J., & Arai, S. (2023). Efficient Elitist Cooperative Evolutionary Algorithm for Multi-Objective Reinforcement Learning. *IEEE Access*, *11*, 43128–43139. https://doi.org/10.1109/access.2023.3272115

**How to cite this Article:**

Leela Bhavani Ranganathan, Archana Rajasundaram and Sasi Kumar Santhosh Kumar (2024). A Cross-Sectional Study on the Effect of Stress on Short-Term Heart Rate Variability and Muscle Strength Among Construction Site Workers. *International Journal of Experimental Research and Review*, *46*, 342-354.

**DOI :** https://doi.org/10.52756/ijerr.2024.v46.027